IN THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A database management system, comprising:

a processor associating multiple different activities with a same transaction, each of the activities consisting of an associated subgroup of program instructions for the same transaction that initiate a subgroup of actions on an associated group of data;

the processor associating separate lock durations with each different subgroup of program instructions associated with the activities in ~~a~~ the transaction, and maintaining locks for the duration of the activities and then releasing the locks when the subgroup of program instructions associated with the activities are completed.

2. (Currently amended) A database management system according to claim 1 wherein one of the activities include a group of individual ~~sort~~ shared lock operations and the processor activates locks for each of the individual ~~sort~~ shared lock operations in the group and releases the locks only when the all of the individual ~~sort~~ shared lock operations in the group are completed.

3. (Original) A database management system according to claim 1 including a memory containing an activity bit map that tracks individual activities for the transaction, the processor assigning activity identifiers to the activities according to the activity bit map.

4. (Original) A database management system according to claim 1 wherein the processor associates the activities with lock modes and releases the lock modes on data items when the associated activities are finished.

5. (Currently amended) A method for locking data items in a database management system, comprising:

associating multiple different activities with a same transaction, each of the activities consisting of an associated subgroup of program instructions for the same transaction initiating a subgroup of actions on an associated group of data;

associating separate individual lock durations with each subgroup of program instructions associated with the different activities in ~~a~~ the transaction;

maintaining locks for the duration of the activities; and

releasing the locks when the subgroup of program instructions associated with the activities are completed.

6. (Currently amended) A method according to claim 5 including:

identifying a plurality of different ~~sort~~ shared operations for the same transaction;

activating locks on data items associated with the plurality of different ~~sort~~ shared operations; and

releasing the locks on the data items only when all of the plurality of different ~~sort~~ shared operations are completed.

7. (Original) A method according to claim 5 including:

maintaining an activity bit map that tracks individual activities for the transaction; and

assigning activity identifiers to the activities according to the activity bit map.

8. (Previously presented) A method according to claim 6 including associating the activities with lock modes and releasing the lock modes on data items when the associated activities are finished.

a processor associating lock durations with different activities in a transaction, and maintaining locks for the duration of the activities and then releasing the locks when the activities are completed.

9. (Previously presented) A method according to claim 1 including:

assigning a same unique activity identifier to multiple different arbitrary database access instructions that constitute the different activities in the transaction, the database access instructions performing one or more operations on multiple data items in a database and the activity identifier assigned to and associated with the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions;

assigning multiple locks to the multiple data items corresponding with the operations performed on the multiple data items pursuant to the database access instructions; and

preventing other transactions and other associated activities from accessing the multiple data items until all of the multiple operations are completed for all of the database access instructions assigned to the activity identifier.

10. (Currently amended) A database management system, comprising:

a processor configured to assign activity identifiers to different individual subgroups of database access instructions for a same transaction that each perform one or more operations on multiple data items in a database, the activity identifiers assigned to and associated with the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions,

the processor further configured to assign multiple locks to the multiple data items corresponding with the operations performed on the multiple data items pursuant to the database access instructions associated with the same activity identifiers and further configured to only release the multiple locks when all of the multiple operations are completed for all of the database access instructions assigned to the same activity identifiers.

11. (Previously presented) The database management system according to claim 10 wherein the processor is further configured to assign the activity identifiers to an arbitrary group of related database access instructions performing operations on an arbitrarily related group of data items.

12. (Previously presented) The database management system according to claim 10 wherein the processor is further configured to assign common transaction identifiers to different related groups of database access instructions assigned different activity identifiers and coordinate when the different related groups of database access instructions are allowed to perform operations on the data items.

13. (Currently amended) A database management system according to claim 10 wherein the processor is configured to assign a first transaction identifier to a group of individual ~~sort~~ shared operations and assign locks to the data items associated with the ~~sort~~ shared operations, the processor further configured to hold the locks until all of the individual ~~sort~~ shared operations in the group have been completed.

14. (Currently amended) Computer readable media containing instructions that when executed by a computer, comprise:

assigning activity identifiers to different individual subgroups of database access instructions within a same transaction that perform multiple operations on multiple data items in a database, the activity identifiers assignable to the database access instructions independently of any relationship that may exist between the multiple data items in the database accessed by the database access instructions; ~~and~~

assigning multiple locks to the multiple data items corresponding with the operations performed on the multiple data items by the different subgroups of database access instructions;

assigning the same activity identifiers to the locks that are associated with the same subgroups of database access instructions; and

[only] releasing all of the multiple locks assigned to the same activity identifiers when all of the multiple operations are completed for all of the subgroups of database access instructions assigned to the same activity identifiers.

15. (Previously Presented) The computer readable media according to claim 14 including instructions that when executed assign the activity identifiers to an arbitrary group of related database access instructions performing operations on an arbitrarily related group of data items.

16. (Previously presented) The computer readable media according to claim 14 including instructions that when executed assign common transaction identifiers to different related groups of database access instructions each assigned different activity identifiers and coordinate when the related groups of database access instructions are allowed to perform operations on the data items.

17. (Currently amended) The computer readable media according to claim 14 including instructions that when executed assign a first transaction identifier to a group of individual ~~sort~~ shared operations, assign locks to the data items associated with the ~~sort~~ shared operations, and ~~to~~ hold the locks until all of the individual ~~sort~~ shared operations in the group have been completed.

18. (Currently amended) A database management system, comprising:

means for associating multiple different activities with a same transaction, each of the activities consisting of an associated subgroup of program instructions for the same transaction that initiate a subgroup of actions on an associated group of data;

means for associating separate sets of locks ~~lock durations~~ with the different activities in ~~a~~ the transaction;

means for maintaining the separate sets of locks for the duration of the different activities; and

means for releasing the separate sets of locks when the associated subgroups of program instructions associated with the activities are completed.

19. (Currently amended) The database management system according to claim 18 including:

means for identifying a plurality of different ~~sort~~ activities for the same transaction;

means for activating locks on data items associated with the ~~sort~~ activities; and

means for releasing the locks on the data items when the associated ~~sort~~ activities are completed.

20. (Previously presented) The database management system according to claim 18 including:

means for maintaining an activity bit map that tracks individual activities for the transaction associated with a same transaction; and

means for assigning activity identifiers to the activities according to the activity bit map.

21. (Previously presented) The database management system according to claim 18 including means for identifying one or more subclasses of activities within an activity and associating lock durations with the subclass and releasing the locks upon completion of the activities in the subclass before releasing the locks on the activity.

22. (New) The database management system according to claim 10 wherein the processor is configured to assign the locks associated transaction identifiers and associated activity identifiers and release groups of the locks only when all of the multiple operations are completed for all of the database access instructions having the same assigned transaction identifiers and activity identifiers.

23. (New) A method, comprising:

assigning a first activity identifier and a transaction identifier to a first group of database access instructions for a transaction;

assigning a first set of locks to a first set of data items accessed by the first group of database access instructions;

identifying a second subset of data items from the first set of data items according to the first group of database access instructions;

releasing the first set of locks when all of the operations for the first group of database access instructions have completed;

assigning a second activity identifier and the same transaction identifier to a second group of database access instructions for the same transaction that modify the second subset of data items identified by the first group of database access instructions;

assigning a second set of locks to the second subset of data items; and

releasing the second set of locks only when all of the operations for the second group of database access instructions have completed modification of the second subset of data items.